7.1.18. The 32-bit LAN Manager networking functions are not supported on Windows 95.

7.1.19. The distributed component object model (DCOM) is not supported on Windows 95.

(This feature is supported on Windows NT 4.0 only.)

7.1.20. Redistributables

Imagehlp is supported on Windows 95 only by shipping Imagehlp.dll as a redistributable. See the Win32 SDK for more information.

OpenGL is supported on Windows 95 only by shipping it as a redistributable. See the Win32 SDK for more information.

7.1.21. Other APIs not supported on Windows 95

AngleArc
PrinterMessageBox
SetColorAdjustment
SetGraphicsMode
ArcTo
ResetPrinter
GetColorAdjustment
CancelDC

7.2. Tips & Tricks for Windows NT 4.0

An online listing of Windows NT 4.0 tips and tricks is available on the WWW at http:\\www.microsoft.com\win32dev\guidelns\testnt40.htm.

7.3. Accessibility

Personal computers are powerful tools that enable people to work, create, and communicate in ways that might otherwise be difficult or impossible. The vision of making computers easier for everyone to use, however, can only be realized if people with disabilities have equal access to personal computing. The Designed for Windows NT and Windows 95 Logo requirements encourage development of products that are accessible to computer users with disabilities. These requirements are based on *The Microsoft Windows Guidelines for Accessible Software Design*.

7.3.1. The need for accessible design

Accessible design is important to all of us in the computer industry for the following reasons:

- It may enable your business customers to comply with the **Americans with Disabilities Act**, which requires that they provide reasonable accommodation for employees with disabilities. Large organizations expect computer applications to accommodate all members of their staff, so failing to meet the needs of a small group of users with disabilities can affect sales to the entire organization.
- It may enable government agencies, schools, and organizations receiving federal funding to comply with section 508 of the **Rehabilitation Act of 1973**, which requires them to purchase office equipment that is usable by people with disabilities.
- It may enable customers to comply with a range of other legal requirements outside of the United States, such as the **Disability Discrimination Act** in the United Kingdom, and upcoming legislation in the European Community.
- It may enable **over 30 million people in the United States alone** who have disabilities to use your products, along with the friends and family members in the households that include people with disabilities.
- It may enable employers to benefit by **keeping valuable employees** even when they receive a permanent or temporary disability (such as repetitive strain injury associated with typing or using a mouse), or experience functional limitations as a natural part of aging.
- It may enable the use of **sophisticated automation tools**, including testing tools, task automation tools such as intelligent agents, and new input methods such as voice input. Such tools function exactly like accessibility aids.
- It may enable your customers to purchase software that meets **standards for usability**, such as HFES/ANSI 200 and ISO 9241. Such standards are currently in draft form and will probably incorporate high-level requirements for accessibility.

- It often makes applications easier for everyone to use. Many people customize their appearance or size schemes, or prefer to keep their hands on the keyboard rather than use a mouse. A flexible user interface may increase user satisfaction and lead to lower fatigue and fewer mistakes. In fact, many technologies that are standard today were originally developed for people with disabilities.
- It may benefit anyone working in a non-optimal circumstances. For example, people may not be able to hear the computer when working in noisy environments such as airplanes, or when the sound is turned off to avoid disturbing other people. Similarly, a person may have difficulty typing in an environment where gloves are required or when one hand is occupied.
- And of course, it is simply **the right thing to do**. When each individual can live, work, study, and play independently, each can make his or her maximum contribution to society.

7.3.2. Summary and explanation of the guidelines

Guidelines in this document address the needs of individuals with the following types of disabilities:

- Blindness
- Low vision and color blindness
- Limited dexterity
- Cognitive and language disabilities (including dyslexia)
- Seizure disorders
- Hard-of-hearing

The following sections list the criteria in this document that affect people with various types of disabilities, with additional explanatory notes. All of these criteria are found in the SHELL/UI or OLE/DCOM sections of this document.

7.3.3. Size

7.3.3.1. Recommended guidelines

- The application should use the system metrics for sizing wherever appropriate.
- Avoid hard coding any font sizes smaller than 10 points.
- Allow the user to choose font names and sizes wherever it is appropriate.

7.3.3.2. Benefits

- **Low Vision**: Individuals with low vision usually require text and objects to be large, or in some cases very large, to be seen clearly. Small fonts cause eyestrain and can make reading difficult or impossible even for people who do not have disabilities.

- **Dexterity**: The larger an object on the screen, the easier it to manipulate. Individuals with impaired dexterity may require objects that are larger than average. (Providing good keyboard access can mitigate these problems.)

7.3.3.3. Notes

Recommended:

The application should be compatible with changes to the system font size and changes to the number of pixels per logical inch.

Notes:

Avoid using the MM_TEXT mapping mode, which is not automatically scaled.

When displaying bitmaps:

- Use StretchBlt to scale images to fit their space in your layout; or
- Clear spaces that may not be entirely filled by their contents, and crop images to fit spaces that are too small.

Verification:

A simple way to verify this is to click the Settings tab on the Display box in Control Panel. Use the Custom Font Size button to select a new logical-to-physical ratio. Then go back into your product and test the major screens, dialog boxes, and controls to ensure that these changes have been properly reflected, that is, they have not created anomalous display situations with wrongly hard-coded elements, or situations where some elements are scaled and others are not.

7.3.4. Color

7.3.4.1. Recommended guidelines

- Support the High Contrast option.
- Do not convey important information by color alone.
- Use system colors wherever appropriate.

7.3.4.2. Benefits

- **Low Vision**: Most users with low vision require specific combinations of foreground and background colors; other colors may be indistinct or painful. Individuals with color blindness have trouble distinguishing between certain colors.
- **Cognitive/Language**: Customizable colors can be used to emphasize certain types of information, such as the focus or selection, which can assist some individuals in focusing.

7.3.4.3. Notes

Recommended:

An application should allow the user to configure all colors, either by using system colors chosen in Control Panel or by providing its own means to configure colors specific to that application. It is recommended that an application use the system colors wherever appropriate, and always in their proper foreground/background combinations.

Notes:

An application can provide its own user interface allowing the user to customize colors specific to the application (for example, in an Options dialog box). Or, at the least, it can support registry keys and provide a .reg file that the user can edit and load to change these settings.

It is strongly recommended that the application use appropriate system colors for any user interface elements that correspond with standard system elements. Examples include window background and window text, button face and button text, and selection foreground and background colors. System colors may be queried using GetSysColor().

System colors must be used in proper combinations. Foreground objects, especially text, must be drawn in foreground system colors, and drawn on the corresponding background color. These colors may be reversed or inverted in specific cases of limited scope or duration, such as when highlighting an object to indicate that it is selected.

The following are examples of **recommended** combinations of foreground and background colors:

- Window text on window background
- Button text on button face
- Highlight text on highlight background
- Menu text on menu background

The following are examples of combinations that are **not recommended** because they mix unrelated colors which are not guaranteed to contrast in all color schemes:

- Window text on button face
- Button text on window background
- Highlight text on window background
- Window text on any hard-coded color

The following are examples of combinations that are **not recommended** because they reverse the meaning of foreground and background:

- Window background on window text
- Button face on button text

Recommended:

Applications should support the High Contrast option, which indicates that the user wants a high degree of legibility.

The High Contrast option is selected through Control Panel and may be queried using SystemParameterInfo to get the value of the SPI_HIGHCONTRAST flag. When it is selected, the application should do the following:

- Omit any background images (watermarks) shown behind text.
- Ensure that foreground and background colors contrast.
- Use only system colors defined by GetSysColor().
- Supplement with text, graphics, or patterns any information that would normally be conveyed by color alone.
- Use monochrome bitmaps if they are available.

If your product allows the user to customize all colors, you should not override the user's choice when the High Contrast option is turned on. If the High Contrast option is on when your product starts, you may want to ask the user whether they want to use the system colors or the colors chosen in your product.

Verification

A simple way to verify proper support for the High Contrast option is to click the **Display** tab on the **Accessibility Options** box in Control Panel. Turn on the High Contrast option, choosing an appearance scheme such as High Contrast Black. Then restart your product, and test the major screens, dialog boxes, and controls, to ensure that these changes have been properly reflected. There should be no anomalous display situations caused by wrongly hard-coded elements. Also, all text and graphic objects should be clearly visible.

7.3.5. Keyboard User Interface

7.3.5.1. Recommended guidelines

Provide keyboard access to all features.

The keyboard user interface should be fully documented.

7.3.5.2. Benefits

Blindness: Many users of blind access utilities find it easier to use the keyboard than a pointing device, especially for moving in free space. Often it is clear to a sighted user how to manipulate objects using the mouse, but it is not clear to those using blind access utilities.

Motor: Many users with impaired dexterity can use the keyboard with software adjustments, but cannot use pointing devices.

7.3.5.3. Notes

The following techniques should be used to ensure that the keyboard user interface is easy to use:

- Model your keyboard user interface on familiar applications or controls. For example, if a custom control is analogous to a push button, make it support the same keystrokes as a standard push button (space, enter). Follow the recommendations in *The Windows Guidelines for User Interface Design.*
- Ensure logical keyboard navigation order. For example, using the tab key in a dialog box should move the focus in an order that seems reasonable, such as left-to-right and top-to-bottom.
- Provide underlined access keys (mnemonics) for all menus and controls, except where this would unavoidably cause conflicts. These may be omitted by default, but should be displayed when the SPI_KEYBOARDPREF option is selected. This option can be determined using SystemParametersInfo().
- Features which require the user to specify a screen location can be handled by providing an alternative interface. For example, a dialog box or property sheet can allow the user to enter coordinates for positioning an object on the screen or on the printed page.
- Do not assume that a keyboard or a mouse will always be the primary input device. Many individuals use speech recognition systems or other alternative input systems, and in some countries handwriting recognition is more efficient than keyboarding.
- Do not assume a standard keyboard layout, as many individuals choose alternative layouts based on their abilities, language, or country. Do not assume that the keyboard has a separate numeric keypad.
- Do not assume that the user is right handed, and therefore that right-handed key combinations are the most convenient.
- When possible, allow the user to customize key assignments.
- Provide keyboard equivalents for drag-and-drop operations.

7.3.6. Exposing Focus and Selection

7.3.6.1. Recommended guidelines

Notify other software of the locations of the keyboard focus and selection.

7.3.6.2. Benefits

Blindness: Blind access utilities need to be able to describe the selection and focus to the user.

Low vision: Screen magnification utilities need to pan to keep the keyboard focus in view.

Dexterity: Alternative input utilities need to know the commands available, which are dependent on the focus context.

Cognitive/Language: Reading aids for people with dyslexia need to read or highlight words with selection or focus.

7.3.6.3. Techniques

Recommended:

Notify other software of the locations of the keyboard focus and selection.

To comply with this recommendation, you should do one of the following:

- Expose objects that have the focus or selection using WinEvents and ActiveX Accessibility. [Note: Beta copies of these components and documentation can be found on http://www.microsoft.com/windows/enable.]
- Move the system caret to track the focus location. Even when the caret is invisible, its location can be tracked by other software. This technique does not allow you to expose the selection, only the keyboard focus location.

Exception:

Objects whose screen location is indicated by their own window do not require any special handling, because the 32-bit Windows operating system sends notification every time the keyboard focus moves to a new window. This only applies to objects which occupy the entire region of their window.

Tip: Indicate the outline of any table cell, custom control, or similar object that receives the keyboard focus.

Indicate the outline of the text caret if you are not using the standard caret functions provided by 32-bit Windows.

7.3.7. Exposing Screen Elements

7.3.7.1. Recommended guidelines

Allow software to identify and manipulate all screen elements that the user interacts with.

7.3.7.2. Benefits

Blindness: Blind access utilities need to be able to describe and activate controls and other screen objects.

Motor: Alternative input utilities need to be able to present the names of commands and activate them when the names are selected.

Cognitive/Language: Voice input utilities need to be able to present the names of commands and activate them when the names are selected; tools for dyslexia need to vocalized or highlighted words on the screen.

7.3.7.3. Notes

Recommended:

The application should allow other software to identify and manipulate all screen elements that the user interacts with.

Verification:

Rather than requiring specific techniques, testing will verify the application's compatibility using testing tools that emulate the behavior of accessibility aids. [This testing tool is currently under development.]

Recommended:

Important screen elements that should be exposed include:

- Windows and window panes
- Controls and hot spots
- Text
- Graphic objects, including regions within larger images

The following are recommended methods of exposing screen elements:

- Use standard windows and controls where possible.
- Use ActiveX Accessibility to expose custom and ActiveX controls.
- Use ActiveX Accessibility and the Text Object Model to expose text fields.

When using standard windows and controls, also follow these guidelines:

- Owner-drawn controls are acceptable if appropriate text is provided.
- Specify user-friendly titles, even if these are not visible.
- Where possible, provide unique window classes for windows that serve different functions.

When you cannot use the techniques described above, the following are acceptable alternatives:

- Use standard GDI calls to draw, copy, and erase text and graphics. Text must be drawn using GDI's text functions rather than relying on bitmaps.
- Label objects with standard tooltip controls.
- Label objects by copying invisible text over them.

7.3.8. Timings

7.3.8.1. Recommended guidelines

Allow the user to customize all user interface timings.

7.3.8.2. Benefits

Blindness: Blind individuals often respond more slowly because of delays imposed by their accessibility aids.

Low Vision: Individuals with low vision often respond more slowly because they can only see a small portion of the screen at a time, so it may take them longer to see or read prompts.

Cognitive/Language: Individuals with cognitive or language disabilities may read slowly, so it may take them longer to read prompts; others may take longer in determining the correct response.

7.3.8.3. Notes

Recommended:

Allow the user to customize all user interface timings that are not based on standard system metrics.

Tips:

Examples of features that use timings include automatic time-outs, auto-scrolling rates, and delay before displaying details of a selected object.

Some users are sensitive to visual information that alternates its appearance or flashes at particular rates—often the greater the frequency, the greater the problem especially in the 10 - 25 Hz range. However, there is nor perfect flash rate. Therefore, base all modulating interfaces on the system's cursor blink rate. Because users can customize this value, a particular frequency can be avoided. If that is not practical, provide your own interface for changing the flash rate.

GetCaretBlinkTime() provides access to the current cursor blink rate setting. For more information about this function, see the documentation included in the Win32 SDK.

If you cannot provide a user interface to allow the user to customize timings, support the option using a registry key and provide a .reg file that can be customized to adjust this setting.

7.3.9. Unexpected Side Effects

7.3.9.1. Recommended guidelines

Don't trigger unexpected side effects based on changes in pointer or keyboard focus locations.

7.3.9.2. Benefits

Blindness: Because blind users cannot perceive the entire screen at once, unexpected events on the screen may not be noticed or may be disorienting.

Low Vision: Because users relying on screen enlargers cannot perceive the entire screen at once, unexpected events on the screen may not be noticed or may be disorienting.

Dexterity: Many users need to minimize the number of keystrokes they type, so unexpected side effects to keyboard navigation can add to that number—sometimes considerably—and be a source of additional pain and frustration.

Cognitive/Language: Unexpected side effects may be confusing to users with some types of cognitive disabilities.

7.3.9.3. Notes

Recommended:

Don't trigger unexpected side effects based on changes in pointer or keyboard focus locations.

Notes:

Some examples of unacceptable behavior include:

- Displaying a dialog box or other window when the keyboard focus moves to a specific region or control.
- Requiring the user to fill in an appropriate value before allowing them to move the keyboard focus off a control.

Exception:

It is acceptable to update screen elements that display attributes of the object with the keyboard focus. Examples include displaying help text for that object in a status bar, or changing a toolbar button that indicates whether the selected text is bolded or in italics.

It is acceptable to change the appearance of the pointer or display tooltips based on the pointer's location.

It is acceptable for pointer motion to trigger side effects when in drag mode.

7.3.10. Sound

7.3.10.1. Guidelines

Do not convey any important information by sound alone.

7.3.10.2. Benefits

Hard-of-hearing: Individuals who are deaf or hard-of-hearing require information to be displayed visually, even if it would normally be presented by sound alone. They may also want to turn off the sound to avoid disturbing people in their surroundings.

7.3.10.3. Notes

Recommended:

Do not convey any important information by sound alone.

Exception:

Game products are exempt from the Sound guidelines.

Exception:

It is not necessary to augment purely decorative sounds that do not convey important information, or sounds whose meaning is redundant to the visual display.

Recommended:

- It is acceptable to present information by sound alone in the default configuration, but such information must be presented visually when the ShowSounds option is selected. This option may be checked using SystemParametersInfo() with the SPI_SHOWSOUNDS flag. An application an also provide its own option to turn on visual displays.
- Turning on visual displays in response to the ShowSound flag or an option provided by your product should not automatically turn off the sounds. It is recommended, but not required, that applications provide a separate option to turn off their sounds.
- Use appropriate techniques for conveying information visually. For example, video clips should be captioned with text synchronized to the visuals. Brief video clips and audio clips of any length can be accompanied by a separate text display or transcript, which does not need to be synchronized to the audio track.
- Audible alerts can be handled simply by playing these sounds using the PlaySound() function. This is sufficient if the sound event is registered with a user-friendly name, because utilities can display this information to the user when the sound is played. Sounds that convey more complex information can be supplemented by message boxes, flashing icons, text displayed in a status field, or similar means.

7.3.11. ActiveX Accessibility

7.3.11.1. Recommended guidelines

Containers should support ActiveX Accessibility, including the WM_GETOBJECT, IDispatch(), and IAccessible() interfaces.

7.3.11.2. Benefits

The same as listed for "Exposing Screen Elements," paragraph 7.3.7.

7.3.11.3. Notes

For information on ActiveX Accessibility, see http://www.microsoft.com/windows/enable.

7.3.12. For more information

For additional information on accessibility and disabilities, see the following resources.

7.3.12.1. Additional Accessibility Guidelines

The most comprehensive guide to accessible design is *The Microsoft Windows Guidelines for Accessible Software Design*, which is published on the Microsoft Developer Network CD-ROM. It is available for browsing on the World Wide Web:

http://www.microsoft.com/win32dev/guidelns/msdnaces.htm

(for online reading)

http://www.microsoft.com/windows/enable

(for download)

The Microsoft guidelines were based on general guidelines first proposed in the white paper "Making Software More Accessible for People with Disabilities." That paper was prepared by Gregg Vanderheiden of the Trace R&D Center of the University of Wisconsin at Madison under funding from the Information Technology Foundation (formerly ADAPSO Foundation) and the National Institute for Disability and Rehabilitation Research (NIDRR) of the U.S. Department of Education. That paper and similar guidelines for other types of products are available at:

http://trace.wisc.edu

or on the *CO-NET CD*, and in print from:

Trace R&D Center	Voice telephone	(608) 263-2309
S-151 Waisman Center	Text telephone	(608) 263-5408
1500 Highland Avenue	Fax	(608) 262-8848
University of Wisconsin	World Wide Web	http://trace.wisc.edu
Madison, WI 53705-2280		

The Windows Interface Guidelines for Software Design, included in the Win32 SDK and available as a book from Microsoft Press, contains a section on accessible software design.

Writing Accessible HTML Documents, a guide to creating accessible documents for the World Wide Web, is available from the Center for Information Technology Accommodation, General Services Administration, Washington D.C. It can be downloaded from http://www.gsa.gov/coca/.

Unified Web Access Guidelines are also available at the Trace Research and Development Center's site at http://trace.wisc.edu/world/web.

7.3.12.2. General Resources

For more information about Microsoft products and services for people with disabilities, or to obtain a listing of third-party accessibility aids for Windows and Windows NT or a listing of resources available to help produce accessible documentation, see:

http://www.microsoft.com/windows/enable

or contact:

Microsoft Sales Information	Voice telephone:	(800) 426-9400
Center One Microsoft Way	Text telephone:	(800) 892-5234
Redmond, WA 98052-6393	Fax:	(206) 936-7329

The Trace Research and Development Center of the University of Wisconsin at Madison produces a book and a compact disc describing products that help people with disabilities use computers. The book, *Trace Resource Book*, provides descriptions and photographs of about 2,000 products. The compact disc, titled *CO-NET CD*, provides a database of more than 18,000 products and other information for people with disabilities. It is issued twice a year. To obtain these directories, see:

http://trace.wisc.edu

or contact:

Trace R&D Center	Voice telephone	(608) 263-2309
S-151 Waisman Center	Text telephone	(608) 263-5408
1500 Highland Avenue	Fax	(608) 262-8848
University of Wisconsin	World Wide Web	http://trace.wisc.edu

Madison, WI 53705-2280

For general information and recommendations about how computers can help specific users, you should consult a trained evaluator who can best match the user needs with the available solutions. An assistive technology program in your area will provide referrals to programs and services that are available to you. To locate the assistive technology program nearest you, you should contact:

National Information System Center for Developmental Disabilities Benson Building University of South Carolina Columbia, SC 29208	Voice/text telephone Fax	(803) 777-4435 (803) 777-6058

7.3.12.3. Customizing for a Specific Operating System

There are many ways you can customize the appearance and behavior of Windows or Microsoft Windows NT to accommodate varying vision and motor abilities without requiring any additional software or hardware. These customizations include adjusting the operating system's appearance, as well as the behavior of the mouse and keyboard. The specific methods available depend on which operating system you are using. Application notes are available describing the specific methods available for each operating system.

For information related to customizing your operating system for people with disabilities, see the appropriate application note for the following operating systems.

Operating system	Application note
Microsoft Windows, version 3.0	Ww0786.txt
Microsoft Windows, version 3.1	Ww0787.txt
Microsoft Windows for Workgroups, version 3.1	Wg0788.txt
Microsoft Windows NT, versions 3.1 and 3.5	Wn0789.exe
Microsoft Windows 95	Wn1062.exe
Microsoft Windows NT, version 4.0	Ww1279.txt

These application notes are available for downloading from the following network services:

- Microsoft's World Wide Web Site on the Internet, at http://www.microsoft.com/windows/enable.
- Microsoft's Internet servers, ftp.microsoft.com and gopher.microsoft.com, in /softlib/mslfiles
- MSN, The Microsoft Network online service
- CompuServe. type GO MSL

- GEnie
- Microsoft Download Service (MSDL), which you can reach by calling (206) 936-6735 any time except between 1:00 A.M. and 2:30 A.M. Pacific time. MSDL supports 1200, 2400, 9600, 14400, or 28800 baud rates (V.32 and V.42), with 8 data bits, no parity, and 1 stop bit.
- Various user-group bulletin boards (such as the bulletin-board services on the Association of PC User Groups network)
- In /SOFTLIB/MSLFILES on the Internet servers FTP.MICROSOFT.COM, GOPHER.MICROSOFT.COM, and WWW.MICROSOFT.COM.

People within the United States who do not have a modem can order these application notes on disks by calling the Microsoft Sales Information Center at (800) 426-9400 (voice telephone) or (800) 892-5234 (text telephone). In Canada, you can call (905) 568-3503 (voice telephone) or (905) 568-9641 (text telephone).

Note: Customers outside the United States can contact the Microsoft subsidiary in their country to find out about the availability of application notes and other resources in their area.

Part Number 098-65959

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.